

Distributed Online Modified Greedy Algorithm for Networked Storage Operation under Uncertainty

Junjie Qin, Yinlam Chow, Jiyang Yang, and Ram Rajagopal

Abstract—The integration of intermittent and stochastic renewable energy resources requires increased flexibility in the operation of the electric grid. Storage, broadly speaking, provides the flexibility of shifting energy over time; network, on the other hand, provides the flexibility of shifting energy over geographical locations. The optimal control of storage networks in stochastic environments is an important open problem. The key challenge is that, even in small networks, the corresponding constrained stochastic control problems on continuous spaces suffer from curses of dimensionality, and are intractable in general settings. For large networks, no efficient algorithm is known to give optimal or provably near-optimal performance for this problem. This paper provides an efficient algorithm to solve this problem with performance guarantees. We study the operation of storage networks, *i.e.*, a storage system interconnected via a power network. An online algorithm, termed Online Modified Greedy algorithm, is developed for the corresponding constrained stochastic control problem. A sub-optimality bound for the algorithm is derived, and a semidefinite program is constructed to minimize the bound. In many cases, the bound approaches zero so that the algorithm is near-optimal. A task-based distributed implementation of the online algorithm relying only on local information and neighbor communication is then developed based on the alternating direction method of multipliers. Numerical examples verify the established theoretical performance bounds, and demonstrate the scalability of the algorithm.

I. INTRODUCTION

Deep penetration of renewable energy generation is essential to ensure a sustainable future. Renewable energy resources, such as wind and solar, are intrinsically variable. Uncertainties associated with these intermittent and volatile resources pose a significant challenge to their integration into the existing grid infrastructure [1]. More flexibility, especially in shifting energy supply and/or demand across time and network, is desired to cope with the increased uncertainties.

Energy storage provides the functionality of shifting energy across time. A vast array of technologies, such as batteries, flywheels, pumped-hydro, and compressed air energy storages, are available for such a purpose [2], [3]. Furthermore, flexible or controllable demand provides another ubiquitous source of storage. Deferrable loads – including many thermal loads, loads of internet data-centers and loads corresponding to charging electric vehicles (EVs) over certain time interval [4]

– can be interpreted and controlled as *storage of demand* [5]. Other controllable loads which can possibly be shifted to an earlier or later time, such as thermostatically controlled loads (TCLs), may be modeled and controlled as a storage with negative lower bound and positive upper bound on the storage level [6]. These forms of storage enable inter-temporal shifting of excess energy supply and/or demand, and significantly reduce the reserve requirement and thus system costs.

On the other hand, shifting energy across a network, *i.e.*, moving excess energy supply to meet unfulfilled demand among different geographical locations with transmission or distribution lines, can achieve similar effects in reducing the reserve requirement for the system. Thus in practice, it is natural to consider these two effects together. Yet, it remains mathematically challenging to formulate a sound and tractable problem that accounts for these effects in electric grid operations. Specifically, due to the power flow and network constraints, control variables in connected buses are coupled. Due to the storage constraints, control variables in different time periods are coupled as well. On top of that, uncertainties associated with stochastic generation and demand dramatically complicate the problem, because of the large number of recourse stages and the need to account for all probable realizations.

Two categories of approaches have been proposed in the literature. The first category is based on exploiting structures of specific problem instances, usually using dynamic programming. These structural results are valuable in providing insights about the system, and often lead to analytical solution of these problem instances. However, such approaches rely heavily on specific assumptions of the type of storage, the form of the cost function, and the distribution of uncertain parameters. Generalizing results to other specifications and more complex settings is usually difficult, and consequently this approach is mostly used to analyze single storage systems. For instance, analytical solutions to optimal storage arbitrage with stochastic price have been derived in [7] without storage ramping constraints, and in [8] with ramping constraints. Problems of using energy storage to minimize energy imbalance are studied in various contexts; see [9], [10] for reducing reserve energy requirements in power system dispatch, [11], [12] for operating storage co-located with a wind farm, [13], [14] for operating storage co-located with end-user demands, and [15] for storage with demand response. The other category is to use heuristic algorithms, such as Model Predictive Control (MPC) [16] and look-ahead policies [17], to identify sub-optimal storage control rules. Usually based on deterministic (convex) optimization, these approaches can be easily applied to general

This research was supported in part by the Satre Family fellowship, and in part by the Tomkat Center for Sustainable Energy.

J. Qin, Y. Chow and J. Yang are with the Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, 94305 USA, (e-mail: {jqin,ychow,jiyan}@stanford.edu).

R. Rajagopal is with the Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: ramr@stanford.edu).

networks. The major drawback is that these approaches usually do not have any performance guarantee. Consequently, it lacks theoretical justification for their implementation in real systems. Examples of this category can be found in [16] and references therein.

This work aims at designing distributed online deterministic optimizations that solve the stochastic control problem with provable guarantees. It contributes to the existing literature in the following ways. First, we formulate the problem of storage network operation as a stochastic control problem with general cost functions, which encapsulates a variety of problems with different types of storage as well as different uses of storage. Second, we devise an online algorithm for the problem based on the theory of Lyapunov optimization, and prove guarantees for its performance in terms of a bound of its sub-optimality. This converts the “intractable” stochastic control program to a sequence of tractable deterministic optimization programs. The bound is useful not only in assessing the performance of our algorithm, but also in evaluating the performance of other sub-optimal algorithms when the optimal costs are hard to obtain. It can also be used to estimate the maximum cost reduction that can be achieved by *any* storage operation, thus provides understanding for the limit of a certain storage system. To the best of our knowledge, this is the first algorithm with provable guarantees for the storage operation problem with general electric networks. Finally, we derive task-based distributed implementation of the online algorithm using the alternating direction method of multipliers (ADMM).

The paper generalizes our prior work [18] by modeling a networked storage system, and extending the online control algorithm to the network setting. Preliminary results related to the network setting have been presented in [19]. The online optimization in this paper is different from that in [19], and the sub-optimality bound here is significant superior to the bound for the algorithm proposed in [19]. The aspect of distributed implementation is also new in this paper.

The rest of the paper is organized as follows. Section 2 formulates the problem of operating a storage network under uncertainty. Section 3 gives the online algorithm and states the performance guarantee. Section 4 discusses the distributed implementation of the online program. Numerical examples are then given in Section 5. Section 6 concludes the paper.

II. PROBLEM FORMULATION

A. Centralized Problem

We model the power grid as a directed graph $G(V, E)$, with $V = [n] \triangleq \{1, \dots, n\}$, $E = [m] \triangleq \{1, \dots, m\}$, where n is the number of nodes and m is the number of edges. The node-edge incidence matrix $A \in \mathbb{R}^{n \times m}$ defined as

$$A_{i,e} = \begin{cases} 1 & \text{if } e \rightarrow i, \\ -1 & \text{if } e \leftarrow i, \\ 0 & \text{otherwise,} \end{cases}$$

where $e \rightarrow i$ denotes that i is the head of e , and $e \leftarrow i$ denotes that i is the tail of e .¹ Here each node models a bus and each edge models a line. To simplify the exposition, we assume each bus i is connected to all of the following types of devices:²

- Uncontrollable net supply. A renewable generator and a load are connected to the bus, with the net power supply, *i.e.*, the generation minus the demand, at time period t denoted by $\delta_i(t)$. As both demand and generation can be stochastic, $\delta_i(t)$ is in general stochastic.
- Energy storage. A storage with storage capacity S_i^{\max} , minimum storage level S_i^{\min} , storage charging limit U_i^{\max} , and storage discharging limit $-U_i^{\min}$ is connected to the bus. The storage level (or state of charge) is denoted by $s_i(t)$ and the storage control is denoted by $u_i(t)$ with $u_i(t) > 0$ representing charging and $u_i(t) < 0$ representing discharging. For each time period t , we have constraints $S_i^{\min} \leq s_i(t) \leq S_i^{\max}$ and $U_i^{\min} \leq u_i(t) \leq U_i^{\max}$. The storage dynamics is $s_i(t+1) = \lambda_i s_i(t) + u_i(t)$, where $\lambda_i \in (0, 1]$ is the storage efficiency which models the energy loss over time without storage operation. We denote the set of parameters for the storage at bus i by $\mathbb{S}_i \triangleq \{\lambda_i, S_i^{\min}, S_i^{\max}, U_i^{\min}, U_i^{\max}\}$. Here the set of parameters for each storage satisfies feasibility and controllability assumptions (see Assumption 2 in Appendix B and [18] for more discussions).
- Conventional generator. Its generation at time period t is denoted by $r_i^+(t) (\geq 0)$ and its convex cost function is denoted by $g_i^+(\cdot)$. It is possible in certain scenarios to have more supply than demand (*e.g.*, when there is too much wind generation). In such cases, let $r_i^-(t) (\geq 0)$ be the generation curtailment at time period t and $g_i^-(\cdot)$ be the cost associated with the curtailment. Without loss of optimality, we can summarize $r_i^+(t)$ and $r_i^-(t)$ by a single variable $r_i(t)$ such that $r_i^+(t) = (r_i(t))^+$ and $r_i^-(t) = (r_i(t))^-$. Then the total cost at bus i and in time period t is

$$g_i(r_i(t)) = g_i^+(r_i^+(t)) + g_i^-(r_i^-(t)).$$

Optionally, the cost can depend on a stochastic price parameter $p_i(t) \in [P_i^{\min}, P_i^{\max}]$, so that we write the cost as $g_i(r_i(t); p_i(t))$.

We use the classic DC approximation for AC power flow. For time period t , let the voltage phase angle on bus i be $\theta_i(t)$. Then the real power flow from bus i to bus j can be written

¹Notation: For a directed graph $G(V, E)$, define $V(e) \triangleq \{i \in V : i \sim e\}$, and $E(i) \triangleq \{e \in E : e \sim i\}$, where $i \sim e$ (and $e \sim i$) means that edge e and node i are incident. We assume that all these sets are equipped with the natural order. For any vector $v \in \mathbb{R}^d$ and $\mathcal{P} \subseteq [d]$, $v_{\mathcal{P}} \in \mathbb{R}^{|\mathcal{P}|}$ is the sub-vector containing entries of v indexed by set \mathcal{P} . Similarly, for any matrix $M \in \mathbb{R}^{d_1 \times d_2}$, and $\mathcal{P}_1 \subseteq [d_1]$ and $\mathcal{P}_2 \subseteq [d_2]$, $M_{\mathcal{P}_1, \mathcal{P}_2} \in \mathbb{R}^{|\mathcal{P}_1| \times |\mathcal{P}_2|}$ is the sub-matrix containing rows and columns of M indexed by sets \mathcal{P}_1 and \mathcal{P}_2 . For any variable $x \in \mathbb{R}^m$ that is defined for each edge, and if edge $e \in E$ is incident to nodes i and j , we use the notations x_e and x_{ij} interchangeably to refer to the e th element of x . For any $x \in \mathbb{R}$, $(x)^+ \triangleq \max(x, 0)$ and $(x)^- \triangleq (-x)^+$. An extended real value function $g(x)$ with domain $\text{dom } g = \mathcal{C} \subseteq \mathbb{R}^d$ is such that $g(x) = \infty$ if $x \notin \mathcal{C}$.

²By setting the problem data properly, we can model buses which are only connected to a subset of these devices. For example, a generator bus with no renewables and no storage can be modeled by setting $\delta_i(t) = 0$ and $S_i^{\min} = S_i^{\max} = 0$.

as

$$f_{ij}(t) = B_{ij}(\theta_i(t) - \theta_j(t)),$$

where $B \in \mathbb{R}^{n \times n}$ is the imaginary part of the admittance matrix (Y-bus matrix) under DC assumptions, and $f_{ij}(t)$ satisfies line flow constraints $-F_{ij}^{\max} \leq f_{ij}(t) \leq F_{ij}^{\max}$, where $F_{ij}^{\max} = F_{ji}^{\max} > 0$ is the real power flow capacity of the line connecting bus i and bus j . Note that our focus here is to identify the optimal operation of storage systems under uncertainty. A more detailed modeling of the AC power flow and incorporating recent convexification techniques [20], [21] into our algorithm are left for future work.

We can now formulate the problem as a stochastic control problem as follows

$$\text{minimize } (1/T) \mathbb{E} \sum_{t=1}^T \sum_{i=1}^n g_i(r_i(t); p_i(t)) \quad (1a)$$

$$\text{subject to } \delta_i(t) + r_i(t) = u_i(t) + \sum_{j=1}^n f_{ij}(t), \quad (1b)$$

$$s_i(t+1) = \lambda_i s_i(t) + u_i(t), \quad (1c)$$

$$U_i^{\min} \leq u_i(t) \leq U_i^{\max}, \quad (1d)$$

$$S_i^{\min} \leq s_i(t) \leq S_i^{\max}, \quad (1e)$$

$$B_{ij}(\theta_i(t) - \theta_j(t)) = f_{ij}(t), \quad (1f)$$

$$-F_{ij}^{\max} \leq f_{ij}(t) \leq F_{ij}^{\max}, \quad (1g)$$

where T is the total number of time periods under consideration, the expectation is taken over $p_i(t)$ and $\delta_i(t)$, constraints (1b), (1c), (1d) and (1e) hold for all i and t , constraints (1f) and (1g) hold for all i, j and t , and $s_i(1) \in [S_i^{\min}, S_i^{\max}]$ is given for each i . Here the goal is to find an optimal control policy for each time period t which maps the information available up to the time period to the optimal decisions $(u^*(t), r^*(t), \theta^*(t), f^*(t))$.

Albeit the bulk of this paper focuses on the formulation (1), we note that it can be extended in various directions.

Remark 1 (Generalized Storage Model): The storage model described above consider primarily energy storage. But following the development in [18], it is easy to incorporate other type of generalized storage such as deferrable loads as storage of demand, and collections of thermostatically controlled load. In addition, the energy loss during charging/discharging can be modeled with conversion functions. For example, a storage with charging coefficient $\mu^C \in (0, 1]$ and discharging coefficient $\mu^D \in (0, 1]$ can be modeled using charging conversion function $h^C(u) = (1/\mu^C)u$ and discharging conversion function $h^D(u) = \mu^D u$, respectively. See [18] for more details.

Remark 2 (Nonconvex Objective): The assumption that g_i is convex for each $i \in [n]$ is not strictly necessary. See [22] for generalization to general subdifferentiable functions.

Remark 3 (Other Costs and Constraints): Many other costs including operational cost of storage due to charging and discharging, and other constraints including bounds on the generation and phase angles can be added without altering our results and the proofs. In fact, the cost can be a function of the form $g_i(u_i(t), r_i(t), \theta_i(t), \delta_i(t), p_i(t))$.

Our prior work [18] can be viewed as the single bus special case of the problem formulated here. Thus the examples for

different use cases of the storage (e.g., balancing and arbitrage) discussed in [18] can also be encapsulated into our current framework together with a network. The incorporation of the network element allows our methodology to be applied to broader problems such as microgrid management and storage-based real-time regulation for the bulk power grid.

B. Cluster based Distributed Control

Solving problem (1) in a centralized fashion may not be feasible due to concerns regarding privacy, communication, and computation. First of all, specifying the centralized problem (1) requires collection of information about the cost functions and parameters of the devices connected to each of the buses, and the probability distributions of all local stochastic parameters. This process involves agents who own the generators, storages, as well as power consumers who may not be willing to report such data. Even if the data reporting is granted, gathering all these data from nodes of a large power network, and subsequently disseminating the optimal control signal obtained from the centralized solution in real time presents a challenge on the communication system required. The large amount of data that have to be sent to and from the centralized control center may lead to traffic congestions and delays in the data delivery. Finally, granting an adequate communication infrastructure in place, solving the stochastic control problem formulated in (1) over a large network is not tractable due to a lack of practical algorithms, i.e., existing algorithms either do not have any performance guarantee or do not scale gracefully with the number of buses of the system.

A cluster-based control architecture for the future grid is envisioned in [23]. Here we present a first step in achieving such an architecture. In particular, we consider solving the centralized problem (1) with resource clusters. Suppose that the network is partitioned into L clusters. Each cluster C_ℓ consists of a subset of nodes $V_\ell \subset V$ and a subset of lines $E_\ell \subset E$, i.e., $C_\ell \triangleq (V_\ell, E_\ell)$, and is controlled by a cluster controller (CC). The CC for each cluster C_ℓ

- possesses local static information including g_i and S_i for all $i \in V_\ell$, and B_e and F_e^{\max} for all $e \in E_\ell$,
- senses local disturbances $\delta_i(t)$ and $p_i(t)$ for all $i \in V_\ell$ and all t ,
- controls local variables $u_i(t)$, $r_i(t)$ and $\theta_i(t)$ for all $i \in V_\ell$, and $f_e(t)$ for all $e \in E_\ell$ and all t ,
- and communicates with its neighbors C_{N_ℓ} where C_{N_ℓ} is the collection of C_w 's for which there exists $e \in E_\ell$, $i \in V_w$ such that $e \sim i$, or there exists $e \in E_w$, $i \in V_\ell$ such that $e \sim i$.

Here we provide a bird-eye view of our approach for tackling the challenging distributed stochastic control problem which we just formulated. Section III provides an online algorithm that converts the centralized stochastic control program to a sequence of online deterministic optimization. Section IV then presents the decentralization of these online deterministic optimization using the alternating direction method of multipliers (ADMM).

III. ONLINE MODIFIED GREEDY ALGORITHM FOR NETWORKED STORAGE CONTROL

A. Algorithm

We propose a very simple algorithm to solve the centralized problem (1) with performance guarantees. The algorithm, termed the network online modified greedy (OMG) algorithm, is composed of an offline and online phase. Next we describe the input data to the algorithm and each phase.

Input Data. Similar to the single storage online modified greedy (OMG) algorithm [18], for each bus $i \in [n]$, in addition to the storage parameters \mathbb{S}_i and the cost functional form g_i , the algorithm requires two input parameters that are a lower bound, denoted by $\underline{D}g_i$, and an upper bound, denoted by $\overline{D}g_i$, for the subdifferential of the objective function g_i with respect to $u_i(t)$.³

Remark 4 (Distribution-Free Method): As in the single storage case [18], The OMG algorithm is a distribution-free method in the sense that almost no information regarding the joint probability distribution of the stochastic parameters $\delta_i(t)$ and $p_i(t)$ are required. The only exception is when calculating $\overline{D}g_i$ and $\underline{D}g_i$, the support of $p_i(t)$ and $\delta_i(t)$ may be needed. Comparing to the entire distribution functions, it is much easier to estimate the supports of the stochastic parameters from historical data.

Offline Phase. Before running the algorithm, each bus $i \in [n]$ needs to calculate two algorithmic parameters, namely a shift parameter Γ_i and a weight parameter W_i . Any pair (Γ_i, W_i) satisfies the following conditions can be used:

$$\Gamma_i^{\min} \leq \Gamma_i \leq \Gamma_i^{\max}, \quad (2)$$

$$0 < W_i \leq W_i^{\max}, \quad (3)$$

where Γ_i^{\min} , Γ_i^{\max} and W_i^{\max} are functions of the storage parameters \mathbb{S}_i and subdifferential bounds $\underline{D}g_i$ and $\overline{D}g_i$.

It will be clear later that the sub-optimality bound depends on the choice of (Γ_i, W_i) . As in [18], we provide two approaches for selecting these parameters

- The *maximum weight* approach (maxW): Setting $W_i = W_i^{\max}$ reduces the interval in (2) to a singleton ($\Gamma_i^{\min} = \Gamma_i^{\max}$) and hence determines a unique Γ_i .
- The *minimum sub-optimality bound* approach (minS): It turns out that the sub-optimality bound of OMG, as a function of (Γ_i, W_i) 's for all $i \in [n]$, can be minimized using a semidefinite program reformulation. This approach uses the set of (Γ_i, W_i) 's minimizing the sub-optimality bound.

Online Phase. At the beginning of each time period t , the OMG algorithm solves a deterministic optimization as follows

$$\text{minimize } \sum_{i=1}^n (\lambda_i / W_i) (s_i + \Gamma_i) u_i + g_i(r_i; p_i) \quad (4a)$$

$$\text{subject to } U_i^{\min} \leq u_i \leq U_i^{\max}, \quad (4b)$$

$$\delta_i + r_i = u_i + \sum_{j=1}^n f_{ij}, \quad (4c)$$

$$B_{ji}(\theta_j - \theta_i) = f_{ji}, \quad (4d)$$

$$-F_{ji}^{\max} \leq f_{ji} \leq F_{ji}^{\max}. \quad (4e)$$

where the optimization variables are u , r , θ and f , and we have dropped the dependence on t to simplify the notation. This treatment is justified by the fact that (4) does not involve the charging and discharging constraints induced by the storage capacity and storage dynamics, *i.e.*, we have removed constraints (1c) and (1e), which can be alternatively summarized as

$$S_i^{\min} \leq \lambda_i s_i + u_i \leq S_i^{\max}. \quad (5)$$

It will be show later in Appendix B that (5) holds automatically given that the algorithmic parameters of OMG satisfy conditions in (2) and (3).

The optimization is similar to the greedy heuristics which minimize the stagewise cost, *i.e.*, $\sum_{i=1}^n g_i(r_i; p_i)$, subject to constraints of (4) together with constraint (5) in each step. Instead of directly optimizing the cost at the current time period, for each storage, the OMG algorithm optimizes a weighted combination of the stage-wise cost and a linear term of u_i depending on the shifted storage level $s_i + \Gamma_i$. Here the weight parameter W_i decides the importance of the original cost in this weighted combination, while the shift parameter Γ_i defines the shifted state given the original state s_i . Roughly speaking, the shifted state $s_i + \Gamma_i$ belongs to an interval $[S_i^{\min} + \Gamma_i, S_i^{\max} + \Gamma_i]$ which usually contains 0. For fixed $W_i > 0$, if the storage level is relatively high, the shifted state is greater than 0, such that the state-dependent term (*i.e.*, $(\lambda_i / W_i)(s_i + \Gamma_i)u_i$) encourages a negative u_i (discharge) to minimize the weighted sum. As a result, the storage level in the next time period will be brought down. On the other hand, if the storage level is relatively low, the shifted state is smaller than 0, such that the state-dependent term encourages a positive u_i (charge) and consequently the next stage storage level is increased. These two effects together help to hedge against uncertainty by maintaining a storage level somewhere in the middle of the feasible interval. More detailed discussion regarding the design of the modification term in the objective can be found in [18].

B. Performance Guarantees

We provide a stylized analysis for the performance of OMG.

Assumption 1: The following assumptions are in force for the analysis in this section.

- A1 Infinite horizon: The horizon length T approaches to infinity.
- A2 IID disturbance: The disturbance process $\{(\delta(t), p(t)) \in \mathbb{R}^{2n} : t \geq 1\}$ is independent and identically distributed (*i.i.d.*) across t and is supported on a compact set such

³Mathematical expressions for these parameters are relegated to Appendix A.

that $\delta_i(t) \in [\delta_i^{\min}, \delta_i^{\max}]$ and $p_i(t) \in [P_i^{\min}, P_i^{\max}]$ for all $i \in [n]$ and all t . Note that any correlation structure is allowed for variables in the same time period.

A3 Frequent acting: The storage parameters satisfy $U_i^{\max} - U_i^{\min} < S_i^{\max} - S_i^{\min}$ for all $i \in [n]$.

Here **A1** and **A2** are technical assumptions introduced to simplify the exposition. Relaxing **A1** leads to no change in our results except an extra term of $O(1/T)$ in the sub-optimality bound. For large T , this term is negligible. [18] discusses how to reduce **A2**. Under these two assumptions, the storage operation problem can be cast as an infinite horizon average cost stochastic optimal control problem in the following form

$$\text{minimize} \quad \lim_{T \rightarrow \infty} (1/T) \mathbb{E} \left[\sum_{t=1}^T \sum_{i=1}^n g_i(r_i(t); p_i(t)) \right] \quad (6a)$$

$$\text{subject to} \quad (1b), (1c), (1d), (1e), (1f), (1g). \quad (6b)$$

Assumption **A3** states that the range of feasible storage control $U_i^{\max} - U_i^{\min}$ is smaller than the range of storage levels $S_i^{\max} - S_i^{\min}$, i.e., the ramping limits of the storage is relatively small compared to the storage capacity. For any storage system, this assumption is true as long as the length of each time period Δt is made small enough; see [18] for more details.

Define $J(u, r, \theta, f)$ as the total cost of problem (1) induced by the sequence of control $\{(u(t), r(t), \theta(t), f(t)), t \geq 1\}$ and $J^* = J(u^*, r^*, \theta^*, f^*)$ as the minimum cost of the average cost stochastic control problem with $\{(u^*(t), r^*(t), \theta^*(t), f^*(t)), t \geq 1\}$ being the corresponding optimal control sequence. The main results regarding the performance of the OMG algorithm is summarized as follows, whose proof is relegated to Appendix B.

Theorem 1 (Performance): The control sequence $(u^{\text{ol}}, r^{\text{ol}}, \theta^{\text{ol}}, f^{\text{ol}}) \triangleq \{(u^{\text{ol}}(t), r^{\text{ol}}(t), \theta^{\text{ol}}(t), f^{\text{ol}}(t)), t \geq 1\}$ generated by the OMG algorithm is feasible with respect to all constraints of problem (1) and its sub-optimality is bounded by $\sum_{i=1}^n M_i(\Gamma_i)/W_i$, that is

$$J^* \leq J(u^{\text{ol}}, r^{\text{ol}}, \theta^{\text{ol}}, f^{\text{ol}}) \leq J^* + \sum_{i=1}^n M_i(\Gamma_i)/W_i, \quad (7)$$

where

$$\begin{aligned} M_i(\Gamma_i) &= M_i^u(\Gamma_i) + \lambda_i(1 - \lambda_i)M_i^s(\Gamma_i), \\ M_i^u(\Gamma_i) &= \frac{1}{2} \max \left((U_i^{\min} + (1 - \lambda_i)\Gamma_i)^2, (U_i^{\max} + (1 - \lambda_i)\Gamma_i)^2 \right), \\ M_i^s(\Gamma_i) &= \max \left((S_i^{\min} + \Gamma_i)^2, (S_i^{\max} + \Gamma_i)^2 \right). \end{aligned}$$

The theorem above guarantees that the cost of the OMG algorithm is no greater than $J^* + \sum_{i=1}^n M_i(\Gamma_i)/W_i$.

In many cases, we are interested to minimize the sub-optimality bound. This can be cast as the following optimization

$$\begin{aligned} \text{PO: minimize} \quad & \sum_{i=1}^n M_i(\Gamma_i)/W_i \\ \text{subject to} \quad & \Gamma_i^{\min} \leq \Gamma_i \leq \Gamma_i^{\max}, \quad 0 < W_i \leq W_i^{\max}, \end{aligned}$$

where the constraints hold for all $i \in [n]$. Observing that the objective is separable across buses, we can solve this program separately on each bus via a semidefinite program (SDP) as

in the single storage case [18]. Here the SDP is reproduced for completeness.

Lemma 2 (Semidefinite Reformulation of PO): For each $i \in [n]$, let symmetric positive definite matrices $X_i^{\min, u}$, $X_i^{\max, u}$, $X_i^{\min, s}$ and $X_i^{\max, s}$ be defined as follows

$$X_i^{(\cdot), u} = \begin{bmatrix} \eta_i^u & U_i^{(\cdot)} + (1 - \lambda_i)\Gamma_i \\ * & 2W_i \end{bmatrix}, \quad X_i^{(\cdot), s} = \begin{bmatrix} \eta_i^s & S_i^{(\cdot)} + \Gamma_i \\ * & W_i \end{bmatrix},$$

where (\cdot) can be either max or min, and η^u and η^s are auxiliary variables. Then **PO** can be solved via the following semidefinite program

$$\text{minimize} \quad \eta_i^u + \lambda_i(1 - \lambda_i)\eta_i^s \quad (9a)$$

$$\text{subject to} \quad \Gamma_i^{\min} \leq \Gamma_i \leq \Gamma_i^{\max}, \quad 0 < W_i \leq W_i^{\max}, \quad (9b)$$

$$X_i^{\min, u}, X_i^{\max, u}, X_i^{\min, s}, X_i^{\max, s} \succeq 0, \quad (9c)$$

where Γ_i^{\min} and Γ_i^{\max} are linear functions of W_i as defined in (13) and (14).

We close this section by summarizing some of the properties for the sub-optimality bound at each bus i in the next remark; more detailed discussion and examples of the uses of the sub-optimality bound can be found at [18].

Remark 5 (Properties of $M_i(\Gamma_i)/W_i$): The following properties are true for the per bus sub-optimality $M_i(\Gamma_i)/W_i$:

- For ideal storage ($\lambda_i = 1$), $M_i(\Gamma_i)/W_i$ is minimized with $W_i = W_i^{\max}$.
- Let the bound minimizing parameter choice be (Γ_i^*, W_i^*) . Then $M_i(\Gamma_i^*)/W_i^* \rightarrow 0$ if (i) $S_i^{\max} - S_i^{\min} \rightarrow \infty$ while $U_i^{\max} - U_i^{\min}$ is fixed or (ii) $U_i^{\max} - U_i^{\min} \rightarrow 0$ while $S_i^{\max} - S_i^{\min}$ is fixed (which may be the case when the storage is controlled frequently such that the length of each time period $\Delta t \rightarrow 0$). That is, when the storage capacity is much larger than the range of feasible storage control action, the algorithm is optimal.

IV. DISTRIBUTED ONLINE CONTROL VIA ALTERNATING DIRECTION METHOD OF MULTIPLIERS

Results in previous section convert the stochastic control program (1) to a sequence of online deterministic optimization programs. In this section, we take a bottom-up approach in deriving a decentralized solution to (1). In particular, we first reformulate the online program and then apply ADMM to obtain a *fully distributed algorithm* that specifies computation and communication tasks for each bus and each line of the network. We then associate the corresponding tasks to the CC's to which these buses or lines belong. For a survey of ADMM, see [24].

A. Node-Edge Reformulation

In order to obtain a fully distributed algorithm that uses only local computation and neighborhood communication, it is necessary to ensure that all constraints of the optimization program only couple variables controlled by pairs of neighboring node and edge so that all communication can be implemented using simple pairwise messages. To this end, we reformulate the online program (4) by creating local copies of certain variables. In particular, let $x_i \triangleq (u_i, r_i, \theta_i, f_{i,E(i)})^\top$ be the local (primal) variables at node i , and $z_e \triangleq (f_e, \theta_{e,V(e)})^\top$ be

the local (primal) variables at edge e , where $\hat{f}_{i,E(i)} \in \mathbb{R}^{|E(i)|}$ is node i 's local auxiliary copy of edge variable $f_{E(i)}$, and $\hat{\theta}_{e,V(e)} \in \mathbb{R}^2$ is edge e 's local auxiliary copy of node variable $\theta_{V(e)}$. Here we use the notation $\hat{f}_{i,e}$ for $e \in E(i)$ to refer to i 's local copy of variable f_e ; similar notation $\hat{\theta}_{e,i}$ is also used. Then program (4) can be written as

$$\text{minimize} \quad \sum_{i=1}^n q_i(x_i) + \sum_{e=1}^m h_e(z_e) \quad (10a)$$

$$\text{subject to} \quad \hat{f}_{i,E(i)} = f_{E(i)}, \forall i \in [n], \quad (10b)$$

$$\hat{\theta}_{e,V(e)} = \theta_{V(e)}, \forall e \in [m], \quad (10c)$$

where extended real value functions q_i and h_e summarize the separable objective and constraints at node i and edge e , respectively, and are defined as follows

$$q_i(x_i) \triangleq q_i(u_i, r_i, \theta_i, \hat{f}_{i,E(i)}) \triangleq (\lambda_i/W_i)(s_i + \Gamma_i)u_i + g_i(r_i; p_i),$$

with domain $\text{dom } q_i = \{x_i : U_i^{\min} \leq u_i \leq U_i^{\max}, \delta_i + r_i + A_{i,E(i)} \hat{f}_{i,E(i)} = u_i, \theta_i \in \mathbb{R}\}$, and $h_e(z_e) = 0$ with $\text{dom } h_e \triangleq \{z_e : f_e = B_e A_{V(e),e}^T \hat{\theta}_{e,V(e)}, -F_e^{\max} \leq f_e \leq F_e^{\max}\}$. Here constraints (10b) and (10c) ensures that at the solution, these local auxiliary variables must be equal to the corresponding true variables. The (scaled) dual variables⁴ corresponding to constraints (10b) and (10c) are denoted by η_i and ξ_e , respectively. We proceed to state the task-based distributed ADMM. The derivation of the algorithm is relegated to Appendix C.

At each iterate, indexed by k , the following tasks are issued and completed in order:

- $\mathcal{T}_i^{\text{NP},k}$: Each node $i \in [n]$ performs *node primal update*:

$$\begin{aligned} x_i^{k+1} = \underset{x_i}{\text{argmin}} \quad & q_i(x_i) + \frac{\rho}{2} \|\hat{f}_{i,E(i)} - f_{E(i)}^k + \eta_i^k\|_2^2 \\ & + \sum_{e \in E(i)} \frac{\rho}{2} (\hat{\theta}_{e,i}^k - \theta_i + \xi_{e,i}^k)^2, \end{aligned}$$

and then passes a message containing θ_i^{k+1} and $\hat{f}_{i,e}^{k+1}$ to each neighboring edge $e \in E(i)$.

- $\mathcal{T}_e^{\text{EP},k}$: Each edge $e \in [m]$ performs *edge primal update*:

$$\begin{aligned} z_e^{k+1} = \underset{z_e}{\text{argmin}} \quad & h_e(z_e) + \frac{\rho}{2} \|\hat{\theta}_{e,V(e)} - \theta_{V(e)}^{k+1} + \xi_e^k\|_2^2 \\ & + \sum_{i \in V(e)} \frac{\rho}{2} (\hat{f}_{i,e}^{k+1} - f_e + \eta_{i,e}^k)^2, \end{aligned}$$

and then passes a message containing f_e^{k+1} and $\hat{\theta}_{e,i}$ to each neighboring node $i \in V(e)$.

- $\mathcal{T}_i^{\text{ND},k}$: Each node $i \in [n]$ performs *node dual update*:

$$\eta_i^{k+1} = \eta_i^k + \hat{f}_{i,E(i)}^{k+1} - f_{E(i)}^{k+1},$$

and passes a message containing $\eta_{i,e}^{k+1}$ to each neighboring edge $e \in E(i)$.

- $\mathcal{T}_e^{\text{ED},k}$: Each edge $e \in [m]$ performs *edge dual update*:

$$\xi_e^{k+1} = \xi_e^k + \hat{\theta}_{e,V(e)}^{k+1} - \theta_{V(e)}^{k+1},$$

and passes a message containing $\xi_{e,i}^{k+1}$ to each neighboring node $i \in V(e)$.

We summarize the convergence property of the iterates specified above, whose proof is relegated to Appendix C.

⁴See Appendix C for more details.

Lemma 3: The iterates (x^k, z^k) produced by tasks $\mathcal{T}^k = \{\mathcal{T}_{[n]}^{\text{NP},k}, \mathcal{T}_{[m]}^{\text{EP},k}, \mathcal{T}_{[n]}^{\text{ND},k}, \mathcal{T}_{[m]}^{\text{ED},k}\}$ are convergent. Let $x^* \triangleq \lim_{k \rightarrow \infty} x^k$ and $z^* \triangleq \lim_{k \rightarrow \infty} z^k$. Then (x^*, z^*) is primal feasible and achieves the minimum cost of problem (10). Furthermore, the rate of convergence is $O(1/k)$.

Remark 6: Minimum amount of assumptions are required to obtain the convergence results given in Lemma 3. In particular, we do not assume the objective function is strongly convex which is a necessary assumption for standard distributed algorithms based on primal or dual decomposition. Furthermore, the rate of convergence for our algorithm is superior to primal or dual decomposition based algorithms, which usually have a rate of convergence $O(1/\sqrt{k})$.

Remark 7 (Asynchronous Variant): Based on the analysis in [25], one can easily extend the algorithm described above to its asynchronous counterpart with similar convergence guarantees.

B. Cluster-based Implementation

In a cluster-based distributed control environment, each CC is responsible for a subset of resources in the grid. It is not necessary the case that there is a CC for each node and each edge. However, issuing tasks defined for each node and edge to the associated CC would implement our distributed algorithm in a cluster-based control environment. The iterates now have the following form: in order, each CC $\ell \in [L]$ (i) performs $\mathcal{T}_i^{\text{NP},k}$ for all $i \in V_\ell$, (ii) performs $\mathcal{T}_e^{\text{EP},k}$ for all $e \in E_\ell$, (iii) performs $\mathcal{T}_i^{\text{ND},k}$ for all $i \in V_\ell$, and (iv) performs $\mathcal{T}_e^{\text{ED},k}$ for all $e \in E_\ell$. Note that if the source and destination of a message belong to different CCs, instead of direct communications between the node-edge pair, the message is sent from the CC containing the source to the CC containing the destination⁵; if a single CC controls both the source and destination of a message, the corresponding messaging step may be skipped.

V. NUMERICAL TESTS

In this section, we show three sets of numerical tests with different focuses. The first example (Subsection V-A) uses synthetic data that honor the i.i.d. assumption in Section III-B to demonstrate the use of the online algorithm and to show how the sub-optimality bound scales with storage parameters. The second example (Subsection V-B) applies the algorithm on IEEE 14 bus network together with real demand and wind data. The i.i.d. assumption no longer holds in this setup. We also demonstrate the convergence of ADMM in this setting. The last example (Subsection V-C) is constructed in particular to show how the distributed algorithm scales with the number of buses of the system. All examples are implemented and tested using Matlab 2014a on a workstation with AMD Magny Cours 24-Core 2.1 GHz CPU and 96GB RAM.

⁵Recall the setup in Section II-B: each CC ℓ can communicate with its neighbors C_{N_ℓ} where C_{N_ℓ} is the collection of C_w 's for which there exists $e \in E_\ell, i \in V_w$ such that $e \sim i$, or there exists $e \in E_w, i \in V_\ell$ such that $e \sim i$. As all messaging tasks only involve incident node-edge pairs, the communication between these CCs are possible.

A. Star Network

Consider a star network, *i.e.*, a tree with a root node and $(n - 1)$ leaf nodes. With a homogeneous setting, all nodes are connected to identical power system components, and thus we only provide specification for a single bus i . The storage network is operated for the purpose of balancing the demand and supply residual due to forecast errors in the wind power generation. The motivation of this setting in a single storage scenario is discussed in detail in [9]. Let $\delta_i(t)$ models the wind forecast error process for each bus i . We simulate the $\delta_i(t)$ processes by generating Laplace distributed random variables with zero mean and standard deviation 0.149 p.u. as in [9], which are estimated empirically using the NREL dataset. Two cases with different cost functions are considered. In the first case, time homogeneous costs of the form

$$g_i(r_i(t); p_i(t)) = p_i^H(t) (r_i(t))^- , \quad (11)$$

are considered, where $p_i^H(t) \equiv 1$; in the second case, the cost function is modified to has a higher penalty rate during the day

$$g_i(r_i(t)) = p_i(t) (r_i(t))^- = \begin{cases} 3 (r_i(t))^- & t \in \mathcal{T}^{\text{Day}}, \\ (r_i(t))^- & \text{otherwise,} \end{cases}$$

with \mathcal{T}^{Day} is the set of time points during the day (7am to 7pm in our tests), $p_i(t) = 3$ if $t \in \mathcal{T}^{\text{Day}}$, and $p_i(t) = 1$ otherwise. We consider non-idealized storages which are operated frequently such that $\lambda_i = 0.999$ with conversion coefficients being $\mu_i^C = \mu_i^D = 0.995$ (*cf.*, Remark 1). We fix $-U_i^{\min} = U_i^{\max} = (1/10)S_i^{\max}$. We have $n = 5$ and $F_e^{\max} = \sigma_\delta$ for each line $e \in [m]$. The time horizon for the simulation is chosen to be $T = 1000$. Figure 1 shows the percentage cost savings compared to the no storage scenario. Albeit the greedy heuristics have been proved to be the optimal solution for single storage systems in the time homogeneous cost setting in [9], OMG outperforms the greedy heuristics in the case with a network. The improvement over the greedy cost is more significant for the time inhomogeneous case. For both cases, the costs of OMG are close to the upper bounds derived using the sub-optimality bounds of the algorithm.

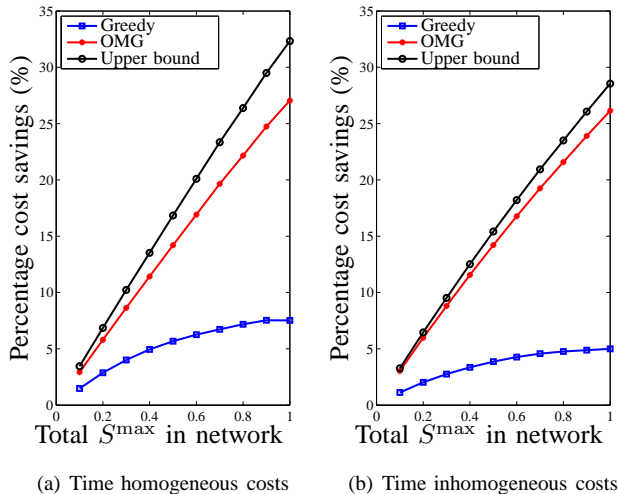


Fig. 1. Percentage cost savings of a storage network operated for balancing.

B. IEEE 14 Bus Case

The network data from IEEE 14 bus test system [26] are used for this example, with modifications described as follows. Three generators are connected to the network, *i.e.*, a coal power plant with capacity 500MW and (constant) marginal generation cost 50\$/MWh connected to bus 1, a nuclear power plant with capacity 450MW and marginal generation cost 25\$/MWh connected to bus 2, and a natural gas power plant with capacity 400MW and marginal generation cost 100\$/MWh is connected to bus 8.⁶ A wind power plant is connected to bus 3. Hourly data of wind power generation for January 2004 (Figure 2) are obtained from the NREL dataset [27], and are scaled to model a 30% penetration scenario. The hourly load data are obtained from PJM interconnection for the same period (Figure 2), and are scaled down and then factored out according to the portion of different load buses. Three storages are connected to buses 6, 7 and 10. Their capacities are $S_6^{\max} = 300\text{MWh}$, $S_7^{\max} = 240\text{MWh}$, $S_{10}^{\max} = 300\text{MWh}$, and charging/discharging power rating are $U_6^{\max} = U_7^{\max} = U_{10}^{\max} = 10\text{MW}$ with $U_i^{\min} = -U_i^{\max}$ for all i . For simplicity (and the fact that conversion to cluster based implementation is easy), we emulate a complete distributed setting, where each node or each edge solves its own tasks in the distributed ADMM algorithm.

The performance of OMG together with the greedy heuristic are simulated over $T = 744$ time periods (*i.e.*, hourly for January 2004). We also compute the cost if there is no storage in the system, and the offline clairvoyant optimal cost which solves the storage operation problem assuming the full knowledge of the future load and wind ahead of time. For this example, the hourly average no storage cost is \$51710. The costs of the greedy heuristics, OMG, and offline optimal are 96.1%, 95.7% and 90.3% of the no storage cost, respectively. Here the cost achieved by the offline optimal solution is a loose lower bound as it requires information that is not available to the decision maker. The stochastic lower bound, estimated by our algorithm under i.i.d. assumption is 94.6% of the no storage cost. As the disturbances are not i.i.d., we expected the actual optimal cost is between these two lower bounds.

The convergence of the fully distributed ADMM is shown in Figure 3. As a comparison, we also plot the convergence of the projected subgradient method (SubGD). Figure 3(a) shows the convergence of the objective value of the online program at a time period for both algorithms with different algorithmic parameter choices, while Figure 3(b) depicts the convergence of the norm of the primal residual for the ADMM algorithm. In terms of the objective value, we observe that the convergence of ADMM is usually much faster compared to SubGD. In fact, in all our examples, SubGD does not converge after thousands of iterations with the tolerance being 1×10^{-4} . Comparing the performance of ADMM with different parameter ρ 's, we note that smaller ρ leads to faster convergence in terms of the objective value but slower convergence of the primal residual. Thus in practice, selecting a ρ that properly trades off these two effects is necessary.

⁶The labeling of the buses are consistent with [26]

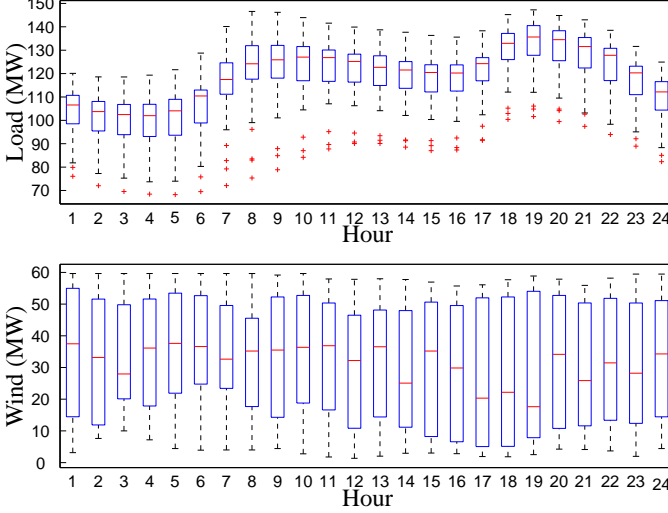
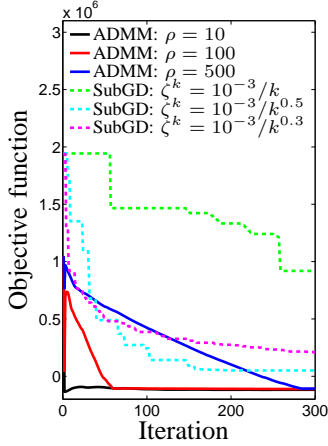
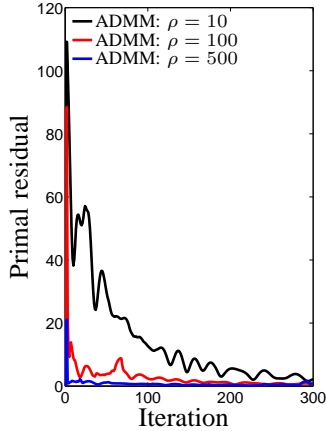


Fig. 2. Bar plots for scaled hourly total load (upper panel) and wind data (lower panel) used for the simulation.



(a) Objective function convergence



(b) ADMM residual convergence

Fig. 3. Convergence of ADMM and centralized subgradient method. Here ζ^k is the step size of the subgradient algorithm at the k th iteration.

C. Scalability

In this subsection, we give a preliminary account for the scalability of the distributed implementation using Matlab Distributed Computing Toolbox. We consider star networks discussed in Subsection V-A with the number of buses increasing from 2 to 16. We associate a processor to each of the buses, and run the distributed ADMM using 2-16 processors. The running time, together with the running time of solving the online programs using the centralized ADMM algorithm, is shown in Figure 4. We note that while in both scenarios, the running time increases approximately linearly with the number of buses, the rate of linear increase for distributed ADMM is significantly smaller. Loading the data for problem specification and communication overheads may have contributed to the linear running time increase for the distributed ADMM.

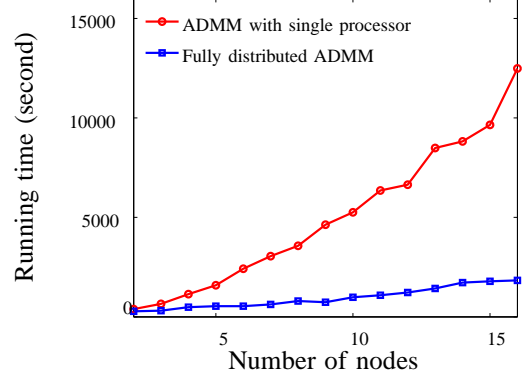


Fig. 4. Running time of distributed and centralized ADMM

VI. CONCLUSION AND FUTURE DIRECTIONS

This paper formulates the storage network operation problem as a stochastic control problem. An online algorithm is proposed to solve the problem efficiently. The performance of the algorithm is analyzed and a sub-optimality bound is derived. The online programs are then solved in a decentralized fashion with only local computation and neighborhood communication with task-based ADMM iterations. Combining these elements, we obtain an efficient task-based distributed online control strategy for operating distributed storage systems with a guaranteed performance.

Many future directions are of interest for generalizing our results. (i) This paper focuses on the real power; incorporating the reactive power and a full AC power flow model may be an important step towards a successful implementation in large-scale practical systems. As the online optimization for each step becomes an AC optimal power flow (OPF) problem, recent work on the convexification of such problems [20] [21], and the distributed solution of the convexified program [28] may be combined with the approach proposed in this paper. (ii) Our decentralized solution is based on the classical two block ADMM which has superior convergence properties compared to other popular methods for distributed optimization such as primal or dual decomposition. Similar methods have

been tested in much larger networks for deterministic energy control problems [29]. However, the fact that such an ADMM algorithm requires a two-block partition (corresponding to the node variables x and edge variables z in Section IV-A) leads to the inconvenience that local copies of variables controlled by the neighbors must be created. Multi-block variants of ADMM may eliminate such need. However, the convergence is not guaranteed or requires additional assumptions [30], [31], [32]. Validating these assumptions for specific storage control problem instances may lead to simpler algorithm which has similar convergence properties. (iii) Utilizing the suboptimality bounds to assess the limit of the storage system for the purpose of storage valuation and system design may also be of interest.

REFERENCES

- [1] National Renewable Energy Laboratory. (2010) Western Wind and Solar Integration Study. [Online]. Available: <http://www.nrel.gov/wind/systemsintegration/wwwis.html>
- [2] —. (2010) The Role of Energy Storage with Renewable Electricity Generation. [Online]. Available: <http://www.nrel.gov/wind/pdfs/47187.pdf>
- [3] D. Lindley, "Smart Grids: The Energy Storage Problem," *Nature*, vol. 463, no. 7277, p. 18, 2010.
- [4] B. Daryanian and R. E. Bohn, "Sizing of Electric Thermal Storage under Real Time Pricing," *IEEE Transactions on Power Systems*, vol. 8, no. 1, pp. 35–43, 1993.
- [5] G. O'Brien and R. Rajagopal, "A Method for Automatically Scheduling Notified Deferrable Loads," in *Proc. of American Control Conference (ACC)*, 2013, pp. 5080–5085.
- [6] H. Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent, "Aggregate Flexibility of Thermostatically Controlled Loads," *IEEE Transactions on Power Systems*, submitted.
- [7] J. Qin, R. Sevlian, D. Varodayan, and R. Rajagopal, "Optimal Electric Energy Storage Operation," in *Proc. of IEEE Power and Energy Society General Meeting*, 2012, pp. 1–6.
- [8] A. Faghih, M. Roozbehani, and M. A. Dahleh, "On the Economic Value and Price-Responsiveness of Ramp-Constrained Storage," *ArXiv e-prints*, 2012. [Online]. Available: arxiv.org/abs/1211.1696
- [9] H. I. Su and A. El Gamal, "Modeling and Analysis of the Role of Energy Storage for Renewable Integration: Power Balancing," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4109–4117, 2013.
- [10] J. Qin, H. I. Su, and R. Rajagopal, "Storage in Risk Limiting Dispatch: Control and Approximation," in *Proc. of American Control Conference (ACC)*, 2013, pp. 4202–4208.
- [11] E. Bitar, R. Rajagopal, P. Khargonekar, and K. Poolla, "The Role of Co-Located Storage for Wind Power Producers in Conventional Electricity Markets," in *Proc. of American Control Conference (ACC)*, 2011, pp. 3886–3891.
- [12] J. H. Kim and W. B. Powell, "Optimal Energy Commitments with Storage and Intermittent Supply," *Operations Research*, vol. 59, no. 6, pp. 1347–1360, 2011. [Online]. Available: <http://pubsonline.informs.org/doi/abs/10.1287/opre.1110.0971>
- [13] P. M. van de Ven, N. Hegde, L. Massoulié, and T. Salonidis, "Optimal Control of End-User Energy Storage," *ArXiv e-prints*, 2012. [Online]. Available: arxiv.org/abs/1203.1891
- [14] R. Uргаonkar, B. Uргаonkar, M. J. Neely, and A. Sivasubramaniam, "Optimal Power Cost Management Using Stored Energy in Data Centers," in *Proc. of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '11, 2011, pp. 221–232.
- [15] L. Huang, J. Walrand, and K. Ramchandran, "Optimal Demand Response with Energy Storage Management," in *Proc. of IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, 2012, pp. 61–66.
- [16] L. Xie, Y. Gu, A. Eskandari, and M. Ehsani, "Fast MPC-Based Coordination of Wind Power and Battery Energy Storage Systems," *Journal of Energy Engineering*, vol. 138, no. 2, pp. 43–53, 2012.
- [17] National Renewable Energy Laboratory. (2013) The Value of Energy Storage for Grid Applications. [Online]. Available: <http://www.nrel.gov/docs/fy13osti/58465.pdf>
- [18] J. Qin, Y. Chow, J. Yang, and R. Rajagopal, "Online Modified Greedy Algorithm for Storage Control under Uncertainty," *ArXiv e-prints*, 2014. [Online]. Available: arxiv.org/abs/1405.7789
- [19] J. Qin, Y. Chow, J. Yang, and R. Rajagopal, "Modeling and Online Control of Generalized Energy Storage Networks," in *Proc. of the 5th International Conference on Future Energy Systems (ACM e-Energy '14)*. ACM, June 2014.
- [20] S. H. Low, "Convex Relaxation of Optimal Power Flow, Part I: Formulations and Equivalence," *ArXiv e-prints*, May 2014. [Online]. Available: arxiv.org/abs/1405.0766
- [21] —, "Convex Relaxation of Optimal Power Flow, Part II: Exactness," *ArXiv e-prints*, May 2014. [Online]. Available: arxiv.org/abs/1405.0814
- [22] J. Qin, Y. Chow, J. Yang, and R. Rajagopal, "Control of Generalized Energy Storage Networks," *Stanford S3L Report*, 2014, available at <http://www.stanford.edu/~jqin/pdf/QCYR2014report.pdf>. [Online]. Available: http://www.stanford.edu/~jqin/pdf/QCYR_2013_report.pdf
- [23] D. Bakken, A. Bose, K. Chandy, P. Khargonekar, A. Kuh, S. Low, A. Von Meier, K. Poolla, P. Varaiya, and F. Wu, "Grip - grids with intelligent periphery: Control architectures for grid2050," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, Oct 2011, pp. 7–12.
- [24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [25] E. Wei and A. Ozdaglar, "On the $O(1/k)$ Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers," *ArXiv e-prints*, 2013. [Online]. Available: arxiv.org/abs/1307.8254
- [26] Power systems test case archive. [Online]. Available: www.ee.washington.edu/research/pstca/
- [27] National Renewable Energy Laboratory, "Eastern wind integration and transmission study," Tech. Rep., 2010.
- [28] E. Dall'Anese, H. Zhu, and G. Giannakis, "Distributed Optimal Power Flow for Smart Microgrids," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1464–1475, Sept 2013.
- [29] M. Krating, E. Chu, J. Lavaei, and S. Boyd, "Dynamic Network Energy Management Via Proximal Message Passing," *Foundations and Trends in Optimization*, vol. 1, no. 2, pp. 1–54, 2013.
- [30] C. Chen, B. S. He, Y. Ye, and X. Yuan, "The Direct Extension of ADMM for Multi-Block Convex Minimization Problems Is Not Necessarily Convergent," *Mathematical Programming*, to appear. [Online]. Available: www.optimization-online.org/DB_FILE/2013/09/4059.pdf
- [31] M. Hong and Z.-Q. Luo, "On the Linear Convergence of the Alternating Direction Method of Multipliers," *arXiv e-print*, 2012. [Online]. Available: arxiv.org/abs/1208.3922
- [32] D. Han and X. Yuan, "A Note on the Alternating Direction Method of Multipliers," *Journal of Optimization Theory and Applications*, vol. 155, no. 1, pp. 227–238, 2012.
- [33] M. J. Neely, "Stochastic Network Optimization with Application to Communication and Queueing Systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.
- [34] B. He and X. Yuan, "On the $O(1/n)$ Convergence Rate of the Douglas–Rachford Alternating Direction Method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.
- [35] M. Hong and Z.-Q. Luo, "On the Linear Convergence of the Alternating Direction Method of Multipliers," *ArXiv e-prints*, Aug. 2012. [Online]. Available: arxiv.org/abs/1208.3922

APPENDIX A

DEFINITIONS AND EXPRESSIONS FOR SECTION III

Here we provide the definition and expressions for $\underline{D}g_i$, $\overline{D}g_i$, Γ_i^{\min} , Γ_i^{\max} and W_i^{\max} . We start by defining $\underline{D}g_i$ and $\overline{D}g_i$ for each $i \in [n]$.

Definition 1: Let $y_i \triangleq (f, \delta_i, p_i)$. For function $\phi_i(u_i, y_i) \triangleq g_i(u_i - \delta_i + \sum_{j=1}^n f_{ij}, p_i)$ that is convex (but not necessarily differentiable) in u_i , a real number α_i is called a (partial) subgradient of ϕ_i with respect to argument u_i at given (u_i, y_i) if $\phi_i(u'_i, y_i) \geq \phi_i(u_i, y_i) + \alpha_i(u'_i - u_i)$ for all $u'_i \in [U_i^{\min}, U_i^{\max}]$. The set of all subgradients at (u_i, y_i) , denoted by $\partial_{u_i} \phi_i(u_i, y_i)$, is called the (partial) subdifferential of $\phi_i(u_i, y_i)$ with respect to u_i at (u_i, y_i) . Denote $\mathcal{U}_i \triangleq$

$[U_i^{\min}, U_i^{\max}]$, $\mathcal{Y}_i \triangleq \mathcal{F} \times [\delta_i^{\min}, \delta_i^{\max}] \times [P_i^{\min}, P_i^{\max}]$ where $\mathcal{F} = \{f : -F_{ij}^{\max} \leq f_{ij} \leq F_{ij}^{\max}, \forall i, j \in [n]\}$. Define the set

$$Dg_i \triangleq \bigcup_{(u_i, y_i) \in \mathcal{U}_i \times \mathcal{Y}_i} \partial_{u_i} \phi_i(u_i, y_i),$$

and let real numbers $\underline{D}g_i$ and $\overline{D}g_i$ be defined such that

$$\underline{D}g_i \leq \inf Dg_i \leq \sup Dg_i \leq \overline{D}g_i. \quad (12)$$

That is, $\underline{D}g_i$ and $\overline{D}g_i$ are a lower bound and an upper bound of the sub-gradient of ϕ_i over its (compact) domain, respectively. More details and examples regarding how to calculate $\underline{D}g_i$ and $\overline{D}g_i$ can be found in our previous work [18]. The bounds for the algorithmic parameters are

$$\Gamma_i^{\min} \triangleq \frac{1}{\lambda_i} (-W_i \underline{D}g_i + U_i^{\max} - S_i^{\max}), \quad (13)$$

$$\Gamma_i^{\max} \triangleq \frac{1}{\lambda_i} (-W_i \overline{D}g_i - S_i^{\min} + U_i^{\min}), \quad (14)$$

and

$$W_i^{\max} \triangleq \frac{(S_i^{\max} - S_i^{\min}) - (U_i^{\max} - U_i^{\min})}{\overline{D}g_i - \underline{D}g_i}. \quad (15)$$

APPENDIX B

PROOF OF THEOREM 1

Similar to the analysis in [18] for the single bus storage case, we will prove Theorem 1 via the following steps:

- 1) Reformulate problem (6) and link it to the sequence of OMG online optimizations (4).
- 2) Prove that the control policy obtained from OMG is feasible to problem (6).
- 3) Derive the performance bound in Theorem 1.

First, we proceed by reformulating problem (6). For $i = 1, \dots, n$, define

$$\bar{u}_i \triangleq \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[\sum_{t=1}^T u_i(t) \right], \quad \bar{s}_i \triangleq \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[\sum_{t=1}^T s_i(t) \right].$$

Note that for $s_i(1) \in [S_i^{\min}, S_i^{\max}]$,

$$\bar{u}_i = \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[\sum_{t=1}^T s_i(t+1) - \lambda_i s_i(t) \right] = (1 - \lambda_i) \bar{s}_i.$$

As $s_i(t) \in [S_i^{\min}, S_i^{\max}]$ for all $t \geq 0$, the above expression implies

$$(1 - \lambda_i) S_i^{\min} \leq \bar{u}_i \leq (1 - \lambda_i) S_i^{\max}.$$

Problem (6) can be equivalently written as follows

$$\mathbf{P1:} \text{ minimize } \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \sum_{t=1}^T \sum_{i=1}^n g_i(r_i(t); p_i(t)) \quad (16a)$$

$$\text{subject to } \delta_i(t) + r_i(t) = u_i(t) + \sum_{j=1}^n f_{ij}(t), \quad (16b)$$

$$s_i(t+1) = \lambda_i s_i(t) + u_i(t), \quad (16c)$$

$$S_i^{\min} - \lambda_i s_i(t) \leq u_i(t) \leq S_i^{\max} - \lambda_i s_i(t), \quad (16d)$$

$$U_i^{\min} \leq u_i(t) \leq U_i^{\max}, \quad (16e)$$

$$(1 - \lambda_i) S_i^{\min} \leq \bar{u}_i \leq (1 - \lambda_i) S_i^{\max} \quad (16f)$$

$$B_{ij}(\theta_i(t) - \theta_j(t)) = f_{ij}(t), \quad (16g)$$

$$-F_{ij}^{\max} \leq f_{ij}(t) \leq F_{ij}^{\max}, \quad (16h)$$

where bounds on $s_i(t)$ are replaced by (16d), and (16f) is added without loss of optimality.

Here we use $J_{P1}(u, r, \theta, f)$ to denote the objective value of **P1** with operation sequence (u, r, θ, f) (as an abbreviation of $\{u(t), r(t), \theta(t), f(t) : t \geq 1\}$), $\Lambda^*(\mathbf{P1}) = (u^*(\mathbf{P1}), r^*(\mathbf{P1}), \theta^*(\mathbf{P1}), f^*(\mathbf{P1}))$ to denote the optimal control sequence for **P1**, $J_{P1}^* \triangleq J_{P1}(\Lambda^*(\mathbf{P1}))$, and we define similar quantities for **P2**. Here **P2** is an auxiliary problem we construct to bridge the infinite horizon storage control problem **P1** to online optimization problems (4). It has the following form

$$\mathbf{P2:} \text{ minimize } \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \sum_{t=1}^T \sum_{i=1}^n g_i(r_i(t); p_i(t)) \quad (17a)$$

$$\text{subject to } \delta_i(t) + r_i(t) = u_i(t) + \sum_{j=1}^n f_{ij}(t), \quad (17b)$$

$$U_i^{\min} \leq u_i(t) \leq U_i^{\max}, \quad (17c)$$

$$(1 - \lambda_i) S_i^{\min} \leq \bar{u}_i \leq (1 - \lambda_i) S_i^{\max} \quad (17d)$$

$$B_{ij}(\theta_i(t) - \theta_j(t)) = f_{ij}(t), \quad (17e)$$

$$-F_{ij}^{\max} \leq f_{ij}(t) \leq F_{ij}^{\max}. \quad (17f)$$

Notice that it has the same objective as **P1**, and evidently it is a relaxation of **P1**. This implies that $u^*(\mathbf{P2})$ may not be feasible for **P1**, and

$$J_{P2}^* = J_{P1}(\Lambda^*(\mathbf{P2})) \leq J_{P1}^*. \quad (18)$$

The reason for the removal of state-dependent constraints (16d) (and hence (16c) as the sequence $\{s(t) : t \geq 1\}$ becomes irrelevant to the optimization of $\{u(t) : t \geq 1\}$) in **P2** is that the state-independent problem **P2** has easy-to-characterize optimal stationary control policies. In particular, from the theory of stochastic network optimization [33], the following result holds.

Lemma 4 (Stationary Disturbance-Only Policies): Under Assumption 1 there exists a stationary disturbance-only policy $\Lambda^{\text{stat}}(t) = (u^{\text{stat}}(t), r^{\text{stat}}(t), \theta^{\text{stat}}(t), f^{\text{stat}}(t))$ satisfying the constraints in **P2** and providing the following guarantees $\forall t$:

$$(1 - \lambda_i) S_i^{\min} \leq \mathbb{E}[u_i^{\text{stat}}(t)] \leq (1 - \lambda_i) S_i^{\max}, \forall i \in [i]$$

$$\mathbb{E} \left[\sum_{i=1}^n g_i(r_i(t); p_i(t)) \middle| \Lambda^{\text{stat}}(t) \right] = J_{P2}^*,$$

where the expectation is taken over the randomization of $\delta_i(t)$, $p_i(t)$, and possibly $\Lambda^{\text{stat}}(t)$ in case the policy is randomized.

Recall the online optimization solved by OMG:

$$\mathbf{P3:} \text{ minimize } \sum_{i=1}^n (\lambda_i / W_i) (s_i + \Gamma_i) u_i + g_i(r_i; p_i) \quad (19a)$$

$$\text{subject to } U_i^{\min} \leq u_i \leq U_i^{\max}, \quad (19b)$$

$$\delta_i + r_i = u_i + \sum_{j=1}^n f_{ij}, \quad (19c)$$

$$B_{ji}(\theta_j - \theta_i) = f_{ji}, \quad (19d)$$

$$-F_{ji}^{\max} \leq f_{ji} \leq F_{ji}^{\max}. \quad (19e)$$

We use $\Lambda^{\text{ol}}(t) = (u^{\text{ol}}(t), r^{\text{ol}}(t), \theta^{\text{ol}}(t), f^{\text{ol}}(t))$ to denote the solution of **P3** at time step t , $\Lambda^*(\mathbf{P3}) = (u^*(\mathbf{P3}), r^*(\mathbf{P3}), \theta^*(\mathbf{P3}), f^*(\mathbf{P3}))$ to denote the sequence $\{\Lambda^{\text{ol}}(t) : t \geq 1\}$, $J_{P3,t}(\Lambda(t))$ to denote the objective function

of **P3** at time period t using policy $\Lambda(t)$, and $J_{\mathbf{P3},t}^*$ to denote the corresponding optimal cost.

Now, we turn to the feasibility analysis of $\Lambda^*(\mathbf{P3})$ with respect to **P1**. Following assumption holds for any storage system that is *controllable*.

Assumption 2 (Feasibility and Controllability): Each storage $i \in [n]$ is feasible and controllable:

- (feasibility) starting from any feasible storage level, there exists a feasible storage operation such that the storage level in the next time period is feasible, i.e., $\lambda_i S_i^{\min} + U_i^{\max} \geq S_i^{\min}$ and $\lambda_i S_i^{\max} + U_i^{\min} \leq S_i^{\max}$.
- (controllability) starting from any feasible storage level, there exists a sequence of feasible storage operations to reach any feasible storage level in a finite number of steps, i.e., $\lambda_i S_i^{\max} + U_i^{\max} \geq S_i^{\max}$ and $\lambda_i S_i^{\min} + U_i^{\min} \leq S_i^{\min}$.

In order to prove that the solution of **P3** is feasible to **P1**, we have the following technical lemma.

Lemma 5: At each time period t , the optimal storage operation of **P3** at node i , $u_i^{\text{ol}}(t)$, for $i = 1, \dots, n$, satisfies

- 1) $u_i^{\text{ol}}(t) = U_i^{\min}$ whenever $\lambda_i \tilde{s}_i(t) \geq -W_i \underline{D}g_i$,
- 2) $u_i^{\text{ol}}(t) = U_i^{\max}$ whenever $\lambda_i \tilde{s}_i(t) \leq -W_i \underline{D}g_i$,

where

$$\tilde{s}_i(t) = s_i(t) + \Gamma_i.$$

Proof: The proof follows from similar arguments used to prove Lemma 3 of [18]. Details are omitted for brevity. ■

We are ready to prove the feasibility of the control sequence generated by the algorithm.

Proof of Theorem 1, feasibility: For any $i = 1, \dots, n$, we first validate that the intervals of Γ_i and W_i are non-empty. By **A3** of Assumption 1, one concludes $W_i^{\max} > 0$, thus it remains to show $\Gamma_i^{\max} \geq \Gamma_i^{\min}$. Based on (15), $W_i \geq 0$, and $\overline{D}g_i \geq \underline{D}g_i$, one obtains

$$W_i(\overline{D}g_i - \underline{D}g_i) \leq [(S_i^{\max} - S_i^{\min}) - (U_i^{\max} - U_i^{\min})].$$

Re-arranging terms results in

$$-W_i \underline{D}g_i + U_i^{\max} - S_i^{\max} \leq -W_i \overline{D}g_i - S_i^{\min} + U_i^{\min},$$

which further implies $\Gamma_i^{\max} \geq \Gamma_i^{\min}$.

We proceed to show that

$$S_i^{\min} \leq s_i(t) \leq S_i^{\max}, \quad (20)$$

for $t = 1, 2, \dots$ and any $i \in [n]$, when $\Lambda^*(\mathbf{P3})$ is implemented. The base case holds by assumption. Let the inductive hypothesis be that (20) holds at time t . The storage level at $t+1$ is then $s_i(t+1) = \lambda_i s_i(t) + u_i^{\text{ol}}(t)$. We show (20) holds at $t+1$ by considering the following three cases.

Case 1. $-W_i \underline{D}g_i \leq \lambda_i \tilde{s}_i(t) \leq \lambda_i (S_i^{\max} + \Gamma_i)$.

First, it is easy to verify that the above interval for $\lambda_i \tilde{s}_i(t)$ is non-empty using (13) and $\Gamma_i \geq \Gamma_i^{\min}$. Next, based on Lemma 5, one obtains $u_i^{\text{ol}}(t) = U_i^{\min} \leq 0$ in this case. Therefore

$$s_i(t+1) = \lambda_i s_i(t) + U_i^{\min} \leq \lambda_i S_i^{\max} + U_i^{\min} \leq S_i^{\max},$$

where the last inequality follows from Assumption 2. On the other hand,

$$\begin{aligned} s_i(t+1) &= \lambda_i s_i(t) + U_i^{\min} \geq -W_i \underline{D}g_i - \lambda_i \Gamma_i + U_i^{\min} \\ &\geq -W_i \underline{D}g_i - \lambda_i \Gamma_i^{\max} + U_i^{\min} \geq S_i^{\min}, \end{aligned}$$

where the third inequality used $\overline{D}g_i \geq \underline{D}g_i$.

Case 2. $\lambda_i (S_i^{\min} + \Gamma_i) \leq \lambda_i \tilde{s}_i(t) \leq -W_i \overline{D}g_i$.

The above interval for $\lambda_i \tilde{s}_i(t)$ is non-empty by (14) and $\Gamma_i \leq \Gamma_i^{\max}$. Lemma 5 implies $u_i^{\text{ol}}(t) = U_i^{\max} \geq 0$ in this case. Therefore, again using Assumption 2,

$$s_i(t+1) = \lambda_i s_i(t) + U_i^{\max} \geq \lambda_i S_i^{\min} + U_i^{\max} \geq S_i^{\min}.$$

On the other hand,

$$\begin{aligned} s_i(t+1) &= \lambda_i s_i(t) + U_i^{\max} \leq -W_i \overline{D}g_i - \lambda_i \Gamma_i + U_i^{\max} \\ &\leq -W_i \overline{D}g_i - \lambda_i \Gamma_i^{\min} + U_i^{\max} \leq S_i^{\max}, \end{aligned}$$

where the third inequality used $\overline{D}g_i \geq \underline{D}g_i$.

Case 3. $-W_i \overline{D}g_i < \lambda_i \tilde{s}_i(t) < -W_i \underline{D}g_i$.

By $U_i^{\min} \leq u_i^{\text{ol}}(t) \leq U_i^{\max}$, one obtains

$$\begin{aligned} s_i(t+1) &= \lambda_i s_i(t) + u_i^{\text{ol}}(t) \leq \lambda_i s_i(t) + U_i^{\max} \\ &< -W_i \underline{D}g_i - \lambda_i \Gamma_i + U_i^{\max} \\ &\leq -W_i \underline{D}g_i - \lambda_i \Gamma_i^{\min} + U_i^{\max} \leq S_i^{\max}. \end{aligned}$$

On the other hand,

$$\begin{aligned} s_i(t+1) &= \lambda_i s_i(t) + u_i^{\text{ol}}(t) \geq \lambda_i s_i(t) + U_i^{\min} \\ &> -W_i \overline{D}g_i - \lambda_i \Gamma_i + U_i^{\min} \\ &\geq -W_i \overline{D}g_i - \lambda_i \Gamma_i^{\max} + U_i^{\min} \geq S_i^{\min}. \end{aligned}$$

Combining these three cases, and by mathematical induction, we conclude (20) holds for all $t = 1, 2, \dots$. ■

It remains to show that the sub-optimality bounds claimed in Theorem 1 indeed hold.

Proof of Theorem 1, performance: Consider a quadratic Lyapunov function $L_i(s_i) = s_i^2/2$. Let the corresponding Lyapunov drift be

$$\Delta_i(s_i(t)) = \mathbb{E}[L_i(s_i(t+1)) - L_i(s_i(t)) | s_i(t)].$$

Recall that $\tilde{s}_i(t+1) = s_i(t+1) + \Gamma_i = \lambda \tilde{s}_i(t) + u_i(t) + (1 - \lambda_i) \Gamma_i$, and so

$$\begin{aligned} \Delta_i(\tilde{s}_i(t)) &= \mathbb{E}[(1/2)(u_i(t) + (1 - \lambda_i) \Gamma_i)^2 - (1/2)(1 - \lambda_i^2) \tilde{s}_i(t)^2 \\ &\quad + \lambda_i \tilde{s}_i(t) u_i(t) + \lambda_i (1 - \lambda_i) \tilde{s}_i(t) \Gamma_i | \tilde{s}_i(t)] \\ &\leq M_i^u(\Gamma_i) - (1/2)(1 - \lambda_i^2) \tilde{s}_i(t)^2 \\ &\quad + \mathbb{E}[\lambda_i \tilde{s}_i(t) u_i(t) + \lambda_i (1 - \lambda_i) \tilde{s}_i(t) \Gamma_i | \tilde{s}_i(t)] \\ &\leq M_i^u(\Gamma_i) + \mathbb{E}[\lambda_i \tilde{s}_i(t) (u_i(t) + (1 - \lambda_i) \Gamma_i) | \tilde{s}_i(t)]. \end{aligned}$$

It follows that, with arbitrary $\Lambda(t) = (u(t), r(t), \theta(t), f(t))$,

$$\begin{aligned} \frac{\Delta_i(\tilde{s}_i(t))}{W_i} + \mathbb{E}[g_i(r_i(t); p_i(t)) | \tilde{s}_i(t)] &\leq \frac{M_i^u(\Gamma_i)}{W_i} + \\ \frac{\lambda_i (1 - \lambda_i) \tilde{s}_i(t) \Gamma_i}{W_i} + \mathbb{E}\left[\frac{\lambda_i \tilde{s}_i(t) u_i(t)}{W_i} + g_i(r_i(t); p_i(t)) | \tilde{s}_i(t)\right]. \end{aligned}$$

By summing the above expression over $i = 1, \dots, n$,

$$\begin{aligned} \sum_{i=1}^n \frac{\Delta_i(\tilde{s}_i(t))}{W_i} + \mathbb{E}[g_i(r_i(t); p_i(t)) | \tilde{s}_i(t)] \\ \leq \sum_{i=1}^n \frac{M_i^u(\Gamma_i)}{W_i} + \frac{\lambda_i (1 - \lambda_i) \tilde{s}_i(t) \Gamma_i}{W_i} + \mathbb{E}[J_{\mathbf{P3},t}(\Lambda(t)) | \tilde{s}(t)]. \end{aligned}$$

where it is clear that minimizing the right hand side of the above inequality over $\Lambda(t)$ is equivalent to minimizing the objective of **P3**. Since $\Lambda^{\text{stat}}(t)$, the disturbance-only stationary

policy of **P2** described in Lemma 4, is feasible for **P3**, then the above inequality implies

$$\begin{aligned}
& \sum_{i=1}^n \frac{\Delta_i(\tilde{s}_i(t))}{W_i} + \mathbb{E}[g_i(r_i(t); p_i(t)) | \tilde{s}_i(t), \Lambda^{\text{ol}}(t)] \\
& \leq \sum_{i=1}^n \frac{M_i^u(\Gamma_i)}{W_i} + \frac{\lambda_i(1-\lambda_i)\tilde{s}_i(t)\Gamma_i}{W_i} + \mathbb{E}[J_{\mathbf{P3},t}^* | \tilde{s}(t)] \\
& \leq \sum_{i=1}^n \frac{M_i^u(\Gamma_i)}{W_i} + \frac{\lambda_i(1-\lambda_i)\tilde{s}_i(t)\Gamma_i}{W_i} + \mathbb{E}[J_{\mathbf{P3},t}(\Lambda^{\text{stat}}(t)) | \tilde{s}(t)] \\
& \stackrel{(a)}{=} \sum_{i=1}^n \frac{M_i^u(\Gamma_i)}{W_i} + \frac{\lambda_i\tilde{s}_i(t)\mathbb{E}[u_i^{\text{stat}}(t) + (1-\lambda_i)\Gamma_i]}{W_i} \\
& \quad + \sum_{i=1}^n \mathbb{E}[g_i(r_i(t); p_i(t)) | \Lambda^{\text{stat}}(t)] \\
& \stackrel{(b)}{\leq} \sum_{i=1}^n \frac{M_i(\Gamma_i)}{W_i} + \mathbb{E}[g_i(r_i(t); p_i(t)) | \Lambda^{\text{stat}}(t)] \\
& \stackrel{(c)}{\leq} \sum_{i=1}^n \frac{M_i(\Gamma_i)}{W_i} + J_{\mathbf{P1}}^*.
\end{aligned} \tag{21}$$

Here (a) uses the fact that $u^{\text{stat}}(t)$ is induced by a disturbance-only stationary policy; (b) follows from inequalities $|\tilde{s}_i(t)| \leq (\max((S_i^{\text{max}} + \Gamma_i)^2, (S_i^{\text{min}} + \Gamma_i)^2))^{1/2}$ and $|\mathbb{E}[u_i^{\text{stat}}(t)] + (1-\lambda_i)\Gamma_i| \leq (1-\lambda_i)(\max((S_i^{\text{max}} + \Gamma_i)^2, (S_i^{\text{min}} + \Gamma_i)^2))^{1/2}$; and (c) used the following equality $\mathbb{E}[\sum_{i=1}^n g_i(r_i(t); p_i(t)) | \Lambda^{\text{stat}}(t)] = J_{\mathbf{P2}}^*$ in Lemma 4 and $J_{\mathbf{P2}}^* \leq J_{\mathbf{P1}}^*$. Taking expectation over $\tilde{s}(t)$ on both sides gives

$$\begin{aligned}
& \mathbb{E} \left[\sum_{i=1}^n g_i(r_i(t); p_i(t)) | \Lambda^{\text{ol}}(t) \right] \\
& + \sum_{i=1}^n \frac{\mathbb{E}[L_i(\tilde{s}_i(t+1)) - L_i(\tilde{s}_i(t))]}{W_i} \leq \sum_{i=1}^n \frac{M_i(\Gamma_i)}{W_i} + J_{\mathbf{P1}}^*.
\end{aligned} \tag{22}$$

Summing expression (22) over t from 1 to T , dividing both sides by T , and taking the limit $T \rightarrow \infty$, we obtain the performance bound in expression (7). ■

APPENDIX C DERIVATION OF THE ADMM ALGORITHM

The first step in deriving the ADMM iterations for the reformulated problem (10) is to form the *augmented Lagrangian function* as follows:

$$\begin{aligned}
& L_\rho(x, z, \mu, \nu) \\
& = \sum_{i=1}^n q_i(x_i) + \mu_i^\top (\hat{f}_{i,E(i)} - f_{E(i)}) + \frac{\rho}{2} \|\hat{f}_{i,E(i)} - f_{E(i)}\|_2^2 \\
& \quad + \sum_{e=1}^m h_e(z_e) + \nu_e^\top (\hat{\theta}_{e,V(e)} - \theta_{V(e)}) + \frac{\rho}{2} \|\hat{\theta}_{e,V(e)} - \theta_{V(e)}\|_2^2,
\end{aligned}$$

where $\mu_i \in \mathbb{R}^{|E(i)|}$ and $\nu_e \in \mathbb{R}^{|V(e)|}$ are dual variables for constraints (10b) and (10c), respectively, and $\rho > 0$ is

a parameter. The centralized ADMM iterates are then

$$x^{k+1} = \underset{x}{\operatorname{argmin}} L_\rho(x, z^k, \mu^k, \nu^k), \tag{23a}$$

$$z^{k+1} = \underset{z}{\operatorname{argmin}} L_\rho(x^{k+1}, z, \mu^k, \nu^k), \tag{23b}$$

$$\mu_i^{k+1} = \mu_i^k + \rho(\hat{f}_{i,E(i)}^{k+1} - f_{E(i)}^{k+1}), \forall i \in [n], \tag{23c}$$

$$\nu_e^{k+1} = \nu_e^k + \rho(\hat{\theta}_{e,V(e)}^{k+1} - \theta_{V(e)}^{k+1}), \forall e \in [m], \tag{23d}$$

where k is the iteration count. Let $\eta_i = \mu_i/\rho$ for all i and $\xi_e = \nu_e/\rho$ for all e be the scaled dual variables. Then upon recognizing that updates (23a) and (23c) are separable across all nodes, and that updates (23b) and (23d) are separable across all edges, we obtain the following distributed ADMM iterates:

$$\begin{aligned}
x_i^{k+1} &= \underset{x_i}{\operatorname{argmin}} q_i(x_i) + \frac{\rho}{2} \|\hat{f}_{i,E(i)} - f_{E(i)}^k + \eta_i^k\|_2^2 \\
& \quad + \sum_{e \in E(i)} \frac{\rho}{2} (\hat{\theta}_{e,i}^k - \theta_i + \xi_{e,i}^k)^2, \forall i \in [n], \\
z_e^{k+1} &= \underset{z_e}{\operatorname{argmin}} h_e(z_e) + \frac{\rho}{2} \|\hat{\theta}_{e,V(e)} - \theta_{V(e)}^{k+1} + \xi_e^k\|_2^2 \\
& \quad + \sum_{i \in V(e)} \frac{\rho}{2} (\hat{f}_{i,e}^{k+1} - f_e + \eta_{i,e}^k)^2, \forall e \in [m], \\
\eta_i^{k+1} &= \eta_i^k + \hat{f}_{i,E(i)}^{k+1} - f_{E(i)}^{k+1}, \forall i \in [n], \\
\xi_e^{k+1} &= \xi_e^k + \hat{\theta}_{e,V(e)}^{k+1} - \theta_{V(e)}^{k+1}, \forall e \in [m].
\end{aligned}$$

The observation that the message passing scheme proposed indeed facilitates the local computation completes this derivation.

Proof of Lemma 3: Based on the derivation above, it is easy to check that the iterations given above implement the standard two block ADMM with x and z being the (two-block) primal variables, and (μ, ν) be the dual variable for the linear equality constraints. The convergence analysis of [24] applies directly. The linear convergence rate follows from e.g. [34] and [35]. ■